



TurboPascal - Prépa HEC Ipecom

Fonctions et procédures 1/2

notes de cours - vendredi 29 mars

1 Intro

Les fonctions et procédures sont des modules, des portions de programme situées en dehors du corps de celui-ci [`begin ...end.`]. Cela permet l'écriture de programmes plus clairs [en décomposant celui-ci en plusieurs modules ayant une fonction précise] et moins répétitifs [les modules peuvent être polyvalents, donc écrits une seule fois pour plusieurs usages].

Une procédure est un module qui prend un ou plusieurs ou aucun arguments, peut les manipuler, afficher des résultats, ..., et renvoie un ou plusieurs ou aucun argument en retour.

Une fonction est un cas particulier, à la syntaxe et à l'usage simplifiés, et qui prend un ou plusieurs arguments, les manipule et renvoie un argument en retour.

Les fonctions et procédures sont déclarées ... dans la partie déclarative du programme, avec les variables, constantes et types. Elles sont utilisées, appelées, invoquées, dans le corps du programme ou dans d'autres fonctions et procédures.

2 Procédure

2.1 Syntaxe simple

```
procedure nom;  
begin  
  ...  
end;
```

REMARQUES:

- Le nom d'une procédure suit les mêmes règles que tout autre nom de variable. On peut définir autant de procédures que nécessaire.
- Dans le corps du programme, ou dans une procédure, l'appel de la procédure `nom_de_proc` se fait simplement par la commande `nom_de_proc;`. L'ensemble des instructions situées entre le `begin` et le `end;` de la déclaration de la procédure sont alors exécutées.

2.2 Variables globales

Les variables globales sont les variables définies et utilisées en-dehors de la procédure [dans la partie déclarative du programme, par exemple]. Ces variables peuvent être utilisées et modifiées dans la procédure, et les modifications effectuées perdurent après l'exécution de la procédure.

2.3 Variables locales

Les variables locales sont des variables déclarées dans une procédure et utilisables uniquement dans celle-ci: on ne peut pas accéder à celles-ci par une instruction qui n'est pas dans le corps de la procédure; si elle a le même nom qu'une variable globale, alors c'est la variable locale qui sera concernée par les instructions de la procédure.

La déclaration a lieu dans la procédure, par une partie déclarative entre l'en-tête de la procédure [`procedure nom;`] et son corps [`begin ...end;`], sous la même forme qu'une déclaration dans le programme :

```
procedure nom;
  var var1 : type1;
      var2,var3 : type23;
begin
  ...
end;
```

On peut également déclarer localement des constantes, des types, etc.

2.4 Arguments par valeur

Les arguments sont des variables locales de la procédure, pour lesquels une valeur est indiquée lors de l'appel de la procédure, ce qui permet plus de souplesse. Lorsqu'ils sont passés par valeur, les arguments ne sont plus accessibles après l'exécution de la procédure.

Ils sont précisés dans la déclaration de la procédure: `procedure nom(var1:type1; var2,var3:type 23...);`

Lors de l'appel de la procédure, on précise quelles valeurs sont envoyées dans quel argument: `nom(1,2,a,b)`, par exemple. L'ordre des arguments et leur nombre doit impérativement correspondre aux arguments déclarés.

2.5 Arguments par adresse

Les arguments passés par adresse sont différents: en fait, le paramètre indiqué lors de l'appel de la procédure n'est pas copié, c'est lui-même qui est manipulé par la procédure; on peut donc observer après l'exécution de la procédure les modifications effectuées.

On les précise également dans la déclaration de la procédure, précédés du mot-clé `var`: `procedure nom(var1:type1; var2,var3:type 23...);`

2.6 Syntaxe complète

```
procedure nom(varv1:typev1; varv2:typev2; ...
              var vara1:typea1; vara2:typea2; ...);
var varl1:type1;
    varl2:typz12;
  ...
begin
  ...
end;
```

3 Fonctions

Une fonction est une procédure simplifiée, d'utilisation un peu plus simple, particulièrement adaptée aux fonctions mathématiques. Une fonction renvoie un seul résultat; le type de celui-ci est indiqué dans la déclaration de la fonction, et ce résultat est manipulé dans le corps de la fonction sous le même nom que cette fonction. Enfin, lors de l'appel de la fonction, l'instruction d'appel est remplacée par la valeur du résultat de la fonction, qui peut être affecté [`a:=nom_de_fonction(1,2);`].

```
function nom(var1 : type1; var2 : type2; ...) : type;
  var var11 : type11;
      var12 : type12;
      ...
begin
  ...
  nom:=...
end;
```

Il est également possible d'utiliser dans une fonction des paramètres passés par adresse, mais cela correspond peu aux besoins dans le cadre d'une fonction, au 'sens' de l'objet fonction.